

# 《The logic route to strong AI》

Alibaba HKAI Lab 2020 presentation

Heelal team

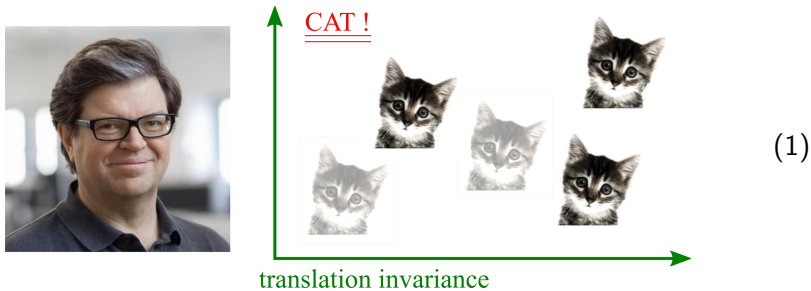
July 24, 2020

# Table of contents

- 1 CNN 在机器视觉中的成功
- 2 Symmetry and inductive bias
- 3 Richard Sutton 的观点
- 4 对 逻辑主义 的质疑
- 5 Structure of logic
- 6 Symmetric neural networks
- 8 应用：知识图谱 (knowledge graphs)
- 9 应用：BERT 的逻辑化
- 10 应用：content-addressable long-term memory
- 11 再谈一次 逻辑结构

# CNN 在机器视觉中的成功

- 在几何学上，视觉 具有 **平移 不变性**：



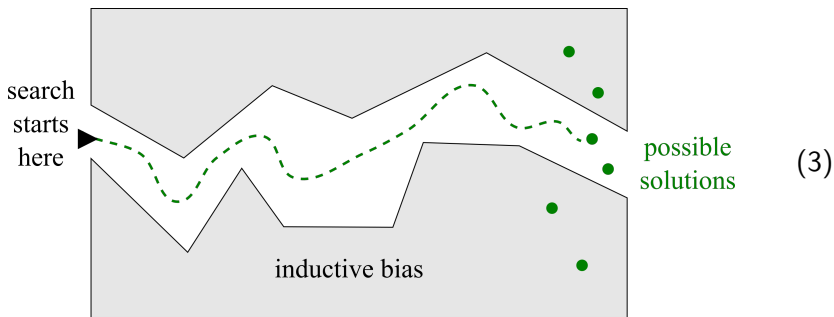
- Convolution 是一种具有平移不变性的运算：

$$(T_x \circ f) * g = T_x \circ (f * g) \quad (2)$$

- Yann LeCun 等人 利用 CNN 的 **对称性** 加快了学习速度，成功地解决了机器视觉 的问题

# Symmetry and inductive bias

- 在数学上, 对称性 经常能简化计算, 所以数学家 特别喜欢 对称
- 在机器学习中, 经常要引入 归纳偏好 (inductive bias), 缩小 搜寻空间:



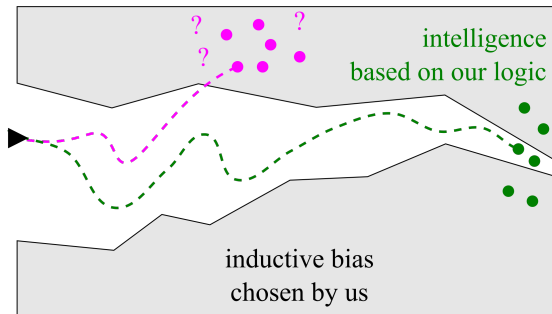
- 往往如果 归纳偏好 选对了, 可以在短时间内找到答案, 否则问题是不可解的 (intractable)

# Richard Sutton 的观点

- Richard Sutton 认为，我们只需在 强化学习 的框架下 增加计算力，就可以找到 strong AI
- 下面描述的只是众多 形式逻辑 之中可能的一种：



intelligence based on "alternative" logics



(4)

- 这不只是一个「空想」的问题；事实上，世界各地的实验室 已经开始了对 AGI 不同形式的搜索！

# 对 逻辑主义 的质疑

- 很多人怀疑：人脑真的用 逻辑 思考吗？
- 其实我们每句表达的 语言，都是逻辑形式的 (logical form)
- 直觉认为，人脑 构造一些 models，再从 model 中「读出」一些结论
- 例如给定一个描述：「已婚妇人出轨，用刀刺死丈夫」



(5)

- 那么 妻子穿著什么衣服？衣服什么颜色？
- 这个 model 可以有哪些细节？答案是：任何细节都不可以有，除非是 逻辑上蕴含的
- 其实人脑可能比我们想像中更接近逻辑化的结构

# Structure of logic

- 我的想法是：在深度学习中引入 **逻辑** 的对称性，解决 strong AI 问题
- 因为人的思维 具有 **逻辑** 的结构，这个 inductive bias 可以帮助我们快速找到 the solution to strong AI
- 逻辑结构很复杂，但最粗略的 symmetry 是 命题的 **可交换律** (commutativity, or permutation invariance):

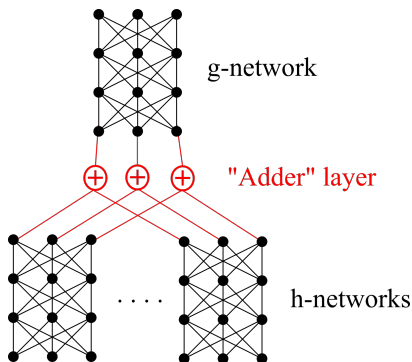
$$\begin{aligned} A \wedge B &\equiv B \wedge A \\ \text{下雨} \wedge \text{失恋} &\equiv \text{失恋} \wedge \text{下雨} \end{aligned} \tag{6}$$

- 它的重要性类似於 视觉中的 平移不变性
- 另一种讲法是：它将智能系统的 **思维状态** (mental state) 分拆成 一粒粒独立的 **命题** (propositions)

# Symmetric neural networks

- Permutation invariance can be handled by **symmetric** neural networks
- 我浪费了两年时间试图解决这问题，却发现现在 3 年前已经有两篇论文解决了 [PointNet 2017] [DeepSets 2017]，而且数学水平比我高很多！
- Any symmetric function can be represented by the following form (a special case of the Kolmogorov-Arnold representation of functions):

$$f(x, y, \dots) = g(h(x) + h(y) + \dots) \quad (7)$$



(8)



- Sym NN gives a powerful boost in efficiency  $\propto n!$  where  $n = \text{\#inputs}$
- The code for Sym NN is just a few lines of Tensorflow:

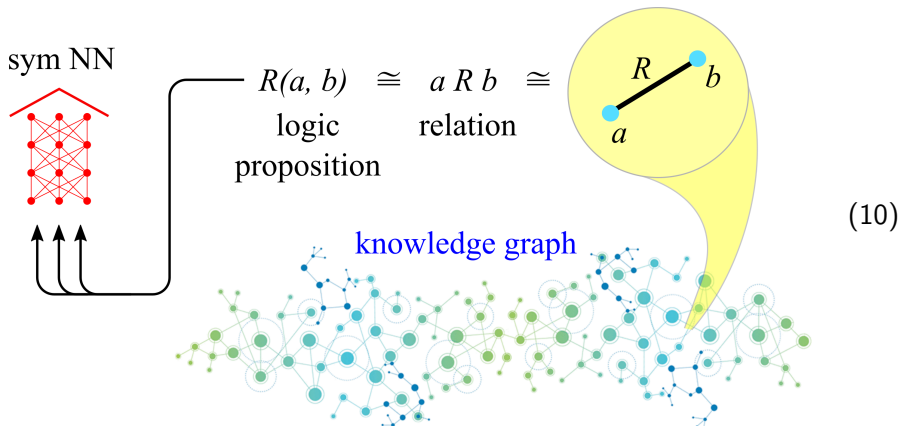
```
h = Dense(3, activation='tanh')
ys = []
for i in range(9):
    ys.append( h(xs[i]) )
y = Keras.stack(ys, axis=1)
Adder = Lambda(lambda x: Keras.sum(x, axis=1))
y = Adder(y)
g = Dense(3)
output = g(y)
```

(9)

- Very easy to adopt this to existing models such as BERT and reinforcement learning
- I have successfully tested it on the game of TicTacToe

## 应用：知识图谱 (knowledge graphs)

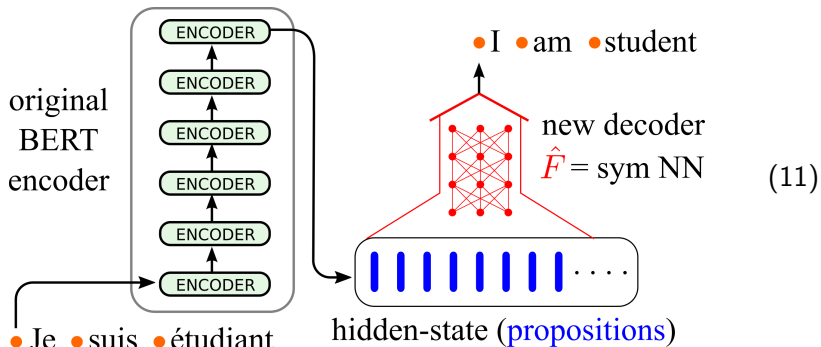
- 知识图谱 不能直接输入神经网络，它必需分拆成很多 edges，每个 edge 是一个 **关系**，也是一个 **逻辑命题**；也可以说 “graphs are isomorphic to logic”



- 而这些 edges 似乎必需用 **symmetric NN** 处理，因为它们是 **permutation invariant**

# 应用：BERT 的逻辑化

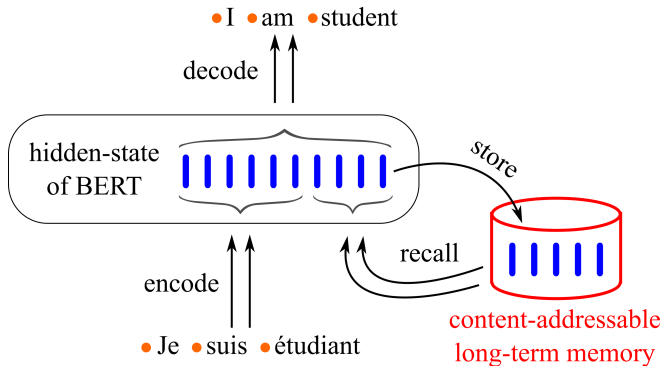
- 类似地，可以将 BERT 的 隐状态 变成 “set of propositions” 的形式，方法是将 原来的 decoder 变成 sym NN：



- 原来的 encoder 可以照旧使用，因为后半部改变了，error propagation 会令 representation 也改变
- 当然，这个想法有待实验证实 😊

## 应用：content-addressable long-term memory

- 以前 BERT 的隐状态 没有逻辑结构，我们不是很清楚它的内容是什么；逻辑化之后，BERT 内部的命题可以储存在 **长期记忆** 中：



(12)

- 这种系统 已非常接近 strong AI，而这是有赖 **逻辑化** 才能做到的
- Content-addressable memory 的想法来自 Alex Graves *et al* 的 Neural Turing Machine [2014]

# 再谈一次 逻辑结构

- 我发现 逻辑 和 AI 之间 有很漂亮的理论
- 现代逻辑理论 揭示出某些 几何 (geometry) 与 拓模 (topology) 结构
- 大家 中 / 小学 时期 都熟悉 Venn diagrams, 其实 命题逻辑 具有 拓扑结构:



而 谓词 (predicates) 是在 base set 上面的 纤维化 (fibration), 记作  $\mathbb{E} \downarrow_P \mathbb{B}$

- 另方面, 著名的 Curry-Howard isomorphism 揭示 逻辑证明 与 编程语言 之间的深刻关系:

$$\begin{aligned} \text{逻辑} &\Leftrightarrow \text{type theory} \\ \text{逻辑证明} &\Leftrightarrow \text{programs} \end{aligned} \quad (14)$$

- 程式 是一些 函数, 而 神经网络 也是非线性的函数映射, 所以 神经网络 也对应于 逻辑推理

- BERT 似乎是在执行 句子之间的变换，而这些句子是 word embedding 的 concatenation，例如：

$$\text{苏格拉底} \cdot \text{是} \cdot \text{人} \xrightarrow{\text{BERT}} \text{苏格拉底} \cdot \text{会} \cdot \text{死} \quad (15)$$

这个做法看似很「粗暴」，其实它和逻辑式子的作用一样：

$$\forall x. \text{Human}(x) \rightarrow \text{Mortal}(x) \quad (16)$$

而这式子，根据 Curry-Howard 对应，就是上面的函数映射

- 这些映射的对象，是形如 “ $a \in A$ ” 这样的物体，它们有 predicates 的几何 / 拓扑结构，但我们现时未做到这一步，暂时只引入了 commutativity
- 从 几何 / 拓扑 “借” 来的概念，演变成 topos 理论、HoTT (homotopy type theory) 等
- 这些漂亮的理论 不是无用的；有需要时，可以從 邏輯 那邊借來更多的結構

# References

- [1] Alex Graves, Greg Wayne, and Ivo Danihelka. “Neural Turing Machines”. In: *CoRR* abs/1410.5401 (2014). arXiv: 1410.5401. URL: <http://arxiv.org/abs/1410.5401>.
- [2] Qi et al. “Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation”. In: *CVPR* (2017). <https://arxiv.org/abs/1612.00593>.
- [3] Zaheer et al. “Deep sets”. In: *Advances in Neural Information Processing Systems* 30 (2017), pp. 3391–3401.

## Illustration credits:

- Translation invariance, from Udacity Course 730, Deep Learning (L3 Convolutional Neural Networks ▷ Convolutional Networks)